Inhalt

[Project Structure 2](#_Toc95405945)

[ERD 2](#_Toc95405946)

[UseCase 2](#_Toc95405947)

[Frontend 3](#_Toc95405948)

[Dependencies 3](#_Toc95405949)

[Architecture 3](#_Toc95405950)

[Introscreen 4](#_Toc95405951)

[Homescreen 4](#_Toc95405952)

[Filter by Categories 5](#_Toc95405953)

[Filter by search (example “co”) 5](#_Toc95405954)

[Product Detail 6](#_Toc95405955)

[Shopping Cart 6](#_Toc95405956)

[Address 7](#_Toc95405957)

[Profile 7](#_Toc95405958)

[Backend 8](#_Toc95405959)

[Datasource config 8](#_Toc95405960)

[Docker setup 8](#_Toc95405961)

[Postman data 8](#_Toc95405962)

[Service testing with Postman 9](#_Toc95405963)

[Add All Categories 9](#_Toc95405964)

[Add All Products 9](#_Toc95405965)

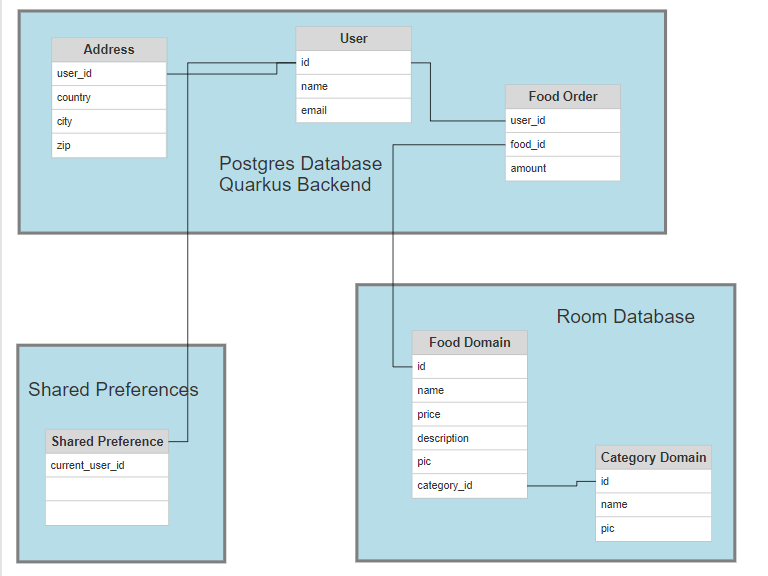
[Get Products by Category 1 (= Pizza-Category) 10](#_Toc95405966)

[Get Food By Id 11](#_Toc95405967)

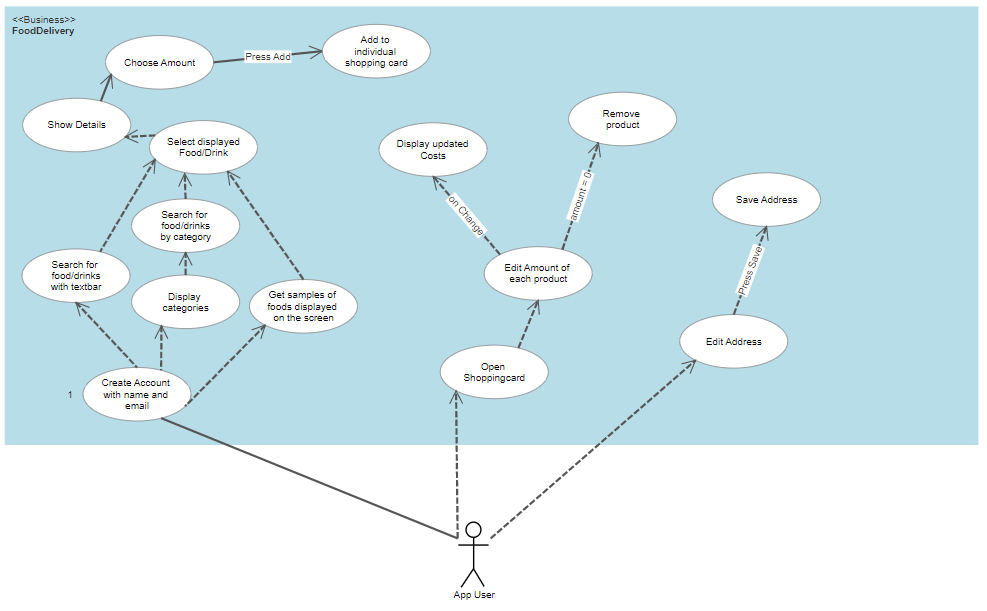
[Get All Categories 12](#_Toc95405968)

# Project Structure

## ERD



## UseCase

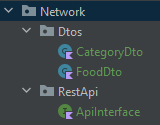


# Frontend

## Dependencies

// retrofit dependencies  
implementation 'com.squareup.retrofit2:retrofit:2.6.1'  
implementation 'com.squareup.retrofit2:converter-gson:2.6.1'  
implementation 'com.jakewharton.picasso:picasso2-okhttp3-downloader:1.1.0'  
  
// Room  
implementation("androidx.room:room-runtime:2.4.1")  
annotationProcessor("androidx.room:room-compiler:2.4.1")  
  
// To use Kotlin annotation processing tool (kapt)  
kapt("androidx.room:room-compiler:2.4.1")

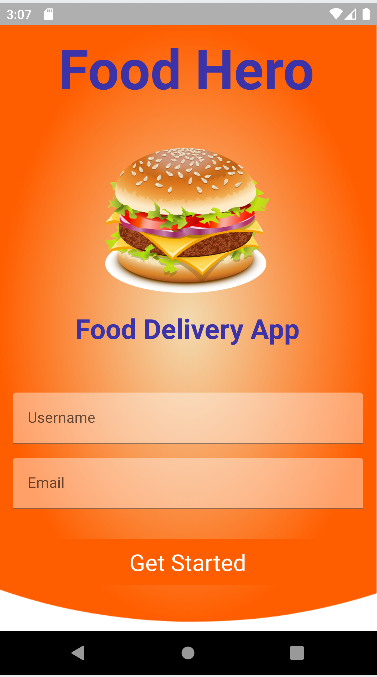
## Architecture

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Introscreen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

After entering the Username and Email, a new Room-User-Entry will be created. Save the input user in the shared preferences Key-Value Pair.   
The Input Email/Name will be controlled before redirect to the Home-Fragment.

## Homescreen

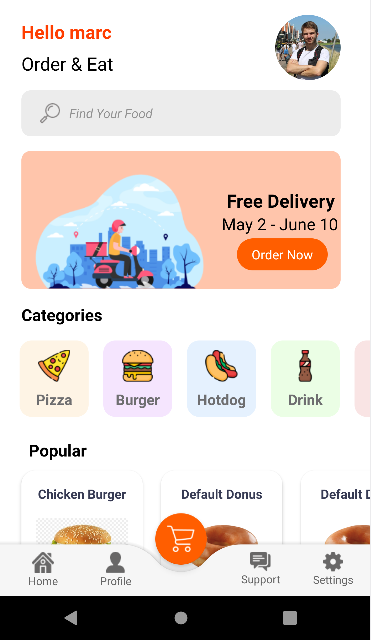
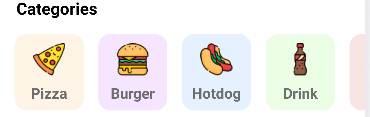
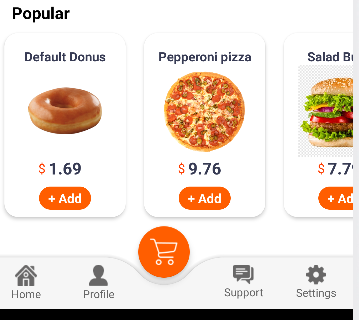


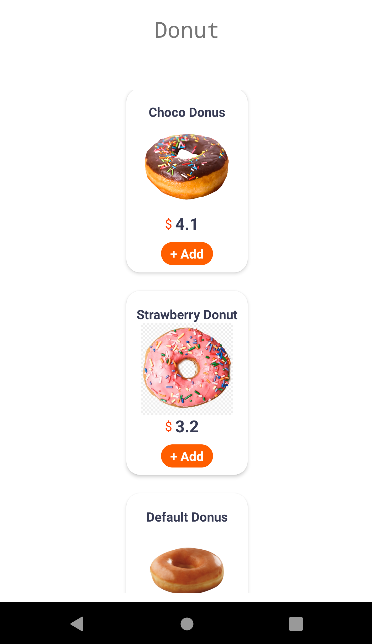
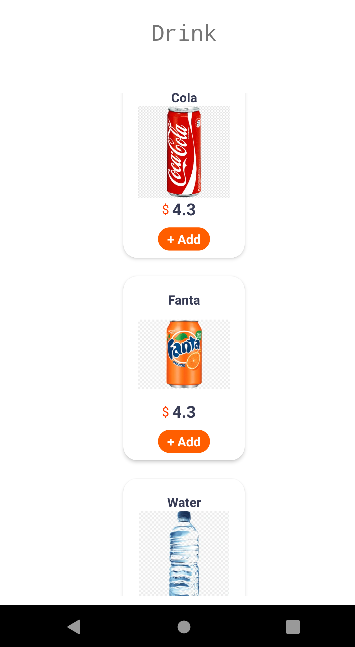
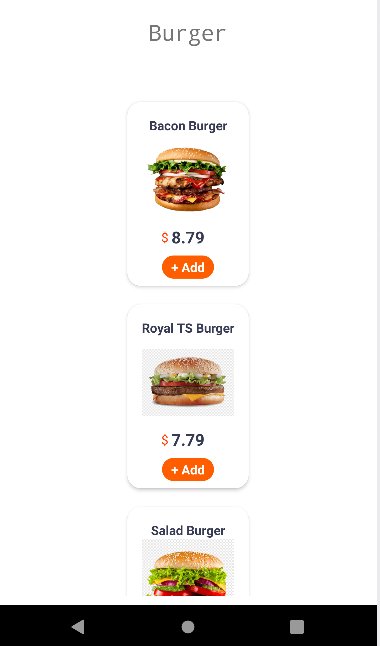
Figure 1 4 Random Sample data from backend

Figure 2 Categories from the backend

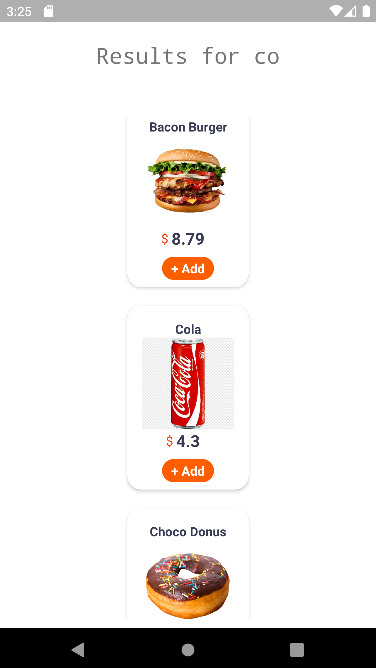
The app loads the Categories and 4 Sample Food-Data from the database and display these things on the home-fragment. The Categories and “Populars” are clickable to redirect to a new page.

Also, the Searchbar can be used to search for a product

## Filter by Categories



## Filter by search (example “co”)



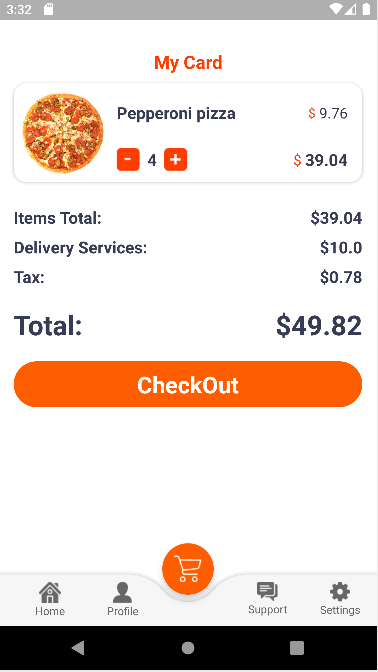
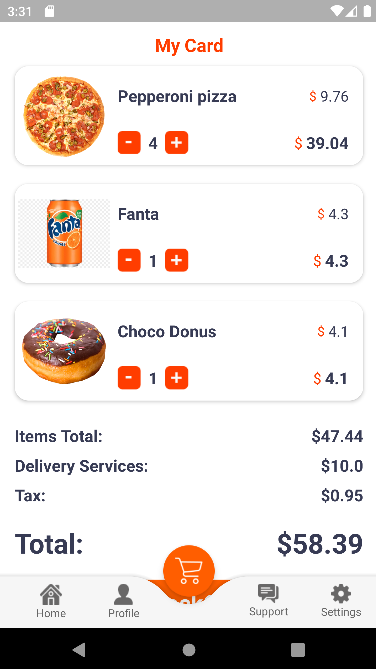
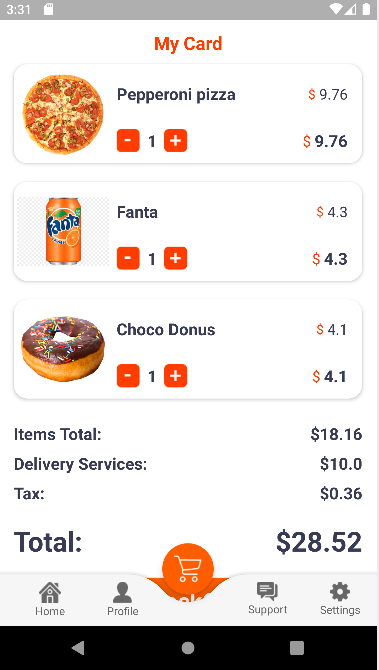
## Product Detail

Ein Bild, das Text, Pizza, Gericht, Essen enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Doughnut, Donut enthält.

Automatisch generierte Beschreibung

## Shopping Cart



These things will be saved, so when you re-log into the app, the shopping card will look the same as when you left it.

## Address

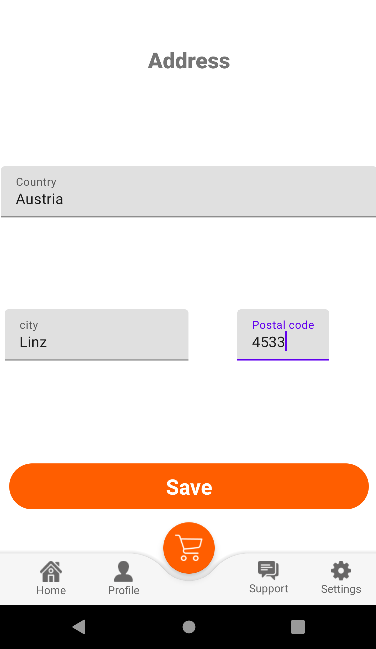




Figure 3 Room Database Address Entry

This Entry can also be updated when changed and saved.

## Profile

In the profile section you can also have a look at your credentials read from the locale room database

Ein Bild, das Text, CD, Screenshot, Vektorgrafiken enthält.

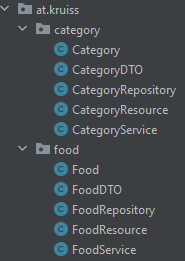
Automatisch generierte Beschreibung

# Backend

## Datasource config

# datasource configuration  
quarkus.datasource.db-kind = postgresql  
quarkus.datasource.username = admin  
quarkus.datasource.password = postgres  
quarkus.datasource.jdbc.url = jdbc:postgresql://localhost:5432/postgres?currentSchema=public  
  
# drop and create the database at startup (use update to only update the schema)  
quarkus.hibernate-orm.database.generation=drop-and-create  
  
quarkus.http.port=8080  
quarkus.http.cors=true

Folder structure

Ein Bild, das Text enthält.

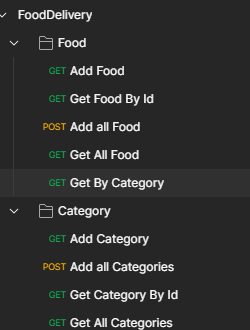
Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung

## Docker setup

docker run --name dbiPostgres -e POSTGRES\_PASSWORD=postgres -e POSTGRES\_USER=admin -d -p 5432:5432 -v dbiPostgres:/var/lib/postgres postgres:alpine

## Postman data



# Service testing with Postman

## Add All Categories

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Monitor, Bildschirm enthält.

Automatisch generierte Beschreibung Figure 4 Response

Figure 5 Add all categories

## Add All Products

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung

Figure 6 Respoonse

Figure 7 Add all products

## Get Products by Category 1 (= Pizza-Category)

Ein Bild, das Text, Screenshot, Monitor, Bildschirm enthält.

Automatisch generierte Beschreibung

[

    {

        "categoryId": 1,

        "description": "slices pepperoni ,mozzarella cheese, fresh oregano,  ground black pepper, pizza sauce",

        "id": 1,

        "name": "Pepperoni pizza",

        "pic": "pizza1",

        "price": 9.76

    },

    {

        "categoryId": 1,

        "description": "Boden aus Teig, Tomatensoße, Käse und weiterer Belag",

        "id": 2,

        "name": "Salami pizza",

        "pic": "pizza3",

        "price": 7.46

    },

    {

        "categoryId": 1,

        "description": "olive oil, Vegetable oil, pitted Kalamata, cherry tomatoes, fresh oregano, basil",

        "id": 3,

        "name": "Vegetable pizza",

        "pic": "pizza2",

        "price": 8.5

    },

    {

        "categoryId": 1,

        "description": "olive oil, Vegetable oil, pitted Kalamata, cherry tomatoes, fresh oregano, basil",

        "id": 4,

        "name": "Vegetable pizza 2",

        "pic": "pizza4",

        "price": 9.5

    },

    {

        "categoryId": 1,

        "description": "olive oil, Vegetable oil, pitted Kalamata, cherry tomatoes, fresh oregano, basil",

        "id": 5,

        "name": "Olives pizza",

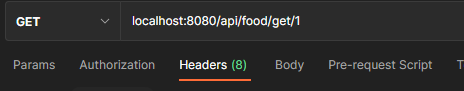
        "pic": "pizza5",

        "price": 11.5

    }

]

## Get Food By Id



{

    "categoryId": 1,

    "description": "slices pepperoni ,mozzarella cheese, fresh oregano,  ground black pepper, pizza sauce",

    "id": 1,

    "name": "Pepperoni pizza",

    "pic": "pizza1",

    "price": 9.76

}

## Get All Categories

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

[

    {

        "id": 21,

        "name": "Pizza",

        "pic": "cat\_1"

    },

    {

        "id": 22,

        "name": "Burger",

        "pic": "cat\_2"

    },

    {

        "id": 23,

        "name": "Hotdog",

        "pic": "cat\_3"

    },

    {

        "id": 24,

        "name": "Drink",

        "pic": "cat\_4"

    },

    {

        "id": 25,

        "name": "Donut",

        "pic": "cat\_5"

    }

]